

AMENDMENTS TO THE CLAIMS

All currently pending claims and status indicators have been reproduced below. This listing will replace all previous versions of the claims.

1. (Currently Amended) An embedded-DRAM (dynamic random access memory) processor comprising:

an embedded DRAM array comprising a plurality of random access memory cells arranged in rows and columns;

a row address register that holds a pointer that points to a row of the DRAM array; one or more sets of registers, each of said sets of registers capable of ~~being loaded~~ loading or ~~stored~~ storing an entire row of the DRAM array in response to a single latch signal; and an instruction set which includes:

- (i) at least one command to perform an arithmetic operation on said row address ~~registers~~ register;
- (ii) a command to precharge (activate) the row pointed to by said row address register;
- (iii) a command to deactivate said row pointed to by said row address register after it had been precharged by the command to precharge;
- (iv) a command to load a plurality of words of the precharged row into designated sets of data registers; and
- (v) a command to load selected columns of the precharged row into designated sets of data registers, said selection based on bits in a mask, wherein all the selected columns of the precharged row are loaded into the designated sets of data registers in a single operation.

2. (Previously Presented) The embedded-DRAM processor according to Claim 1, further comprising:

first and second sets of functional units, said first and second sets of functional units having respective first and second instruction sets and capable of accessing first and second sets of said registers;

a command to select one of said first and second sets of registers to be an architectural set of registers accessible to said first set of functional units;

a command to deselect the other of said first and second sets of registers so that it is no longer an architectural register set accessible to said first set of functional units;

a command to select one of said first and second sets of registers to be an architectural set of registers accessible to said second set of functional units; and

a command to deselect the other one of said first and second sets of registers so that it is no longer an architectural register set accessible to said second set of functional units.

3. (Previously Presented) The embedded-DRAM processor according to Claim 1, further comprising:

first and second sets of functional units, said first and second sets of functional units having respective first and second instruction sets and accessing first and second sets of registers; and

a command which selects one of said first and second sets of registers to be an architectural set of registers accessible to said first set of functional units, and, at the same time, which deselects the other one of said one of said first and second sets of registers to be an architectural set of registers accessible to said second set of functional units.

4. (Previously Presented) The embedded-DRAM processor according to Claim 1, further comprising:

first and second sets of functional units, said first and second sets of functional units having respective first and second instruction sets and accessing first and second ones of said register sets; and

whereby said first and second instruction sets are subsets of said instruction set of said embedded-DRAM processor.

5. (Previously Presented) The embedded-DRAM processor according to Claim 4, whereby said second set of functional units comprises a functional unit that is a multi-issue functional unit, and further comprises:

a dispatch unit;

a plurality of functional units which each execute a respective instruction stream as dispatched by said dispatch unit.

6. (Original) The embedded-DRAM processor according to Claim 1, further comprising a plurality of DRAM arrays.

7. (Previously presented) The embedded-DRAM processor according to Claim 1, further comprising:

at least one functional unit;

whereby said one or more sets of registers comprise a plurality of register files, each of said register files comprising a parallel access port operative to load or store contents of said register file in a single cycle from or to a DRAM row as selected by said row-address register, each of said register files further comprising at least a second access port operative to transfer data between said functional unit and a selected subset register in said register file.

8. (Previously Presented) The embedded-DRAM processor according to Claim 7, further comprising:

a second functional unit;

whereby said first functional unit executes a first command to perform logical processing on the contents of one or more registers within a selected active one of said register sets, and said

second functional unit executes a second command to parallelly transfer data between a selected inactive one of said register sets and said DRAM array.

9. (Previously Presented) The embedded-DRAM processor according to Claim 8, wherein said first and second functional units execute said first and second commands substantially contemporaneously.

10. (Original) The embedded-DRAM processor according to Claim 8, further comprising:
a first software module comprising a set of data manipulation commands, said first software module executed by said first functional unit; and

a second software module comprising a set of parallel data transfer commands, said second software module being executed by said second functional unit;

whereby said second software module operates in support of said first software module to prefetch data from said DRAM array into one of said register files in advance of said data being needed by said first software module.

11. (Previously Presented) The embedded-DRAM processor according to Claim 10, wherein:
said first software module contains an instruction that reference registers within an architectural register set visible to said first functional unit, whereby said architectural register set corresponds at least partially to said one of said register files that is in an active state;
said second software module contains instructions that cause data to be transferred between an inactive register set and said DRAM array, and second software module also executes a command to toggle a selected register set between said active and inactive states.

12. (Previously Presented) The embedded-DRAM processor according to Claim 1, further comprising:

first and second sets of functional units, said first and second sets of functional units having respective first and second instruction subsets;

whereby said first instruction subset includes said command (i) and the second instruction subset includes said commands (ii), (iii), (iv) and (v).

13. (Currently Amended) An embedded-DRAM (dynamic random access memory) processor comprising:

an embedded DRAM array comprising a plurality of random access memory cells arranged in rows and columns;

a row address register that holds a pointer that points to a row of the DRAM array; one or more sets of data registers, each of said sets of data registers capable of ~~being loaded~~ loading or ~~stored~~ storing an entire row of the DRAM array in response to a single latch signal;

a bit mask to select one or more data locations within at least one of said register sets; and an instruction set which comprises at least:

- (i) a command to perform arithmetic operations on said row address register;
- (ii) a command to precharge (activate) the row pointed to by said row address register;
- (iii) a command to load a set of selected elements of the precharged row pointed to by said row address register into a selected set of said data registers, said selection of elements based on bits in said bit mask, wherein all the selected elements of the precharged row are loaded into the selected sets data registers in a single operation;

wherein the command to precharge is executed to precharge the row prior to the command to load so that at the time the command to load is issued, the command to load can execute without the need to wait for the row to precharge.

14. (Previously Presented) The embedded-DRAM processor according to Claim 13, wherein said load command causes an entire row that was previously precharged to be loaded.

15. (Cancelled)

16. (Currently Amended) An embedded-DRAM (dynamic random access memory) processor comprising:

an embedded DRAM array comprising a plurality of random access memory cells arranged in rows and columns;

a row address register that holds a pointer that points to a row of the DRAM array;

first and second register files, each of said register files having a plurality of data registers capable of ~~being loaded~~ loading or ~~stored~~ storing an entire row of the DRAM array in response to a single latch signal, each of said register files also being capable of being placed into an active state and an inactive state;

a set of functional units that perform logical operations on data accessed from a set of architectural registers, wherein registers placed into the active state appear as architectural registers to a set of functional units, and registers in the inactive state are not accessible by the functional units;

a bit mask to select one or more locations within at least one of said register files; and an instruction set which comprises at least:

- (i) a command to perform an arithmetic operation on said row address register; and
- (ii) a command to load a set of selected elements of the row pointed to by said row address register into a selected set of said data registers, said selection of elements based on bits in said bit mask, wherein all the selected elements of the row are loaded into the selected sets data registers in a single operation, and the selected set of said data registers being in the inactive state.

17. (Previously Presented) The embedded-DRAM processor of Claim 16, wherein the instruction set further comprises:

- (iii) a command to toggle a register set between said active and inactive states.

18. (Previously Presented) The embedded-DRAM processor of Claim 17, wherein said toggle command causes said first register tile to toggle from the inactive state to the active state and also causes the second register file to toggle from the active state to the inactive state.

19. (Previously Presented) The embedded-DRAM processor of Claim 16, wherein the instruction set further comprises:

- (iii) a command to manipulate the bits in the bit mask.

20. (Previously Presented) The embedded-DRAM processor according to Claim 16, further comprising:

first and second sets of functional units;

first and second sets of instructions capable of accessing said first and second register sets; and

said instruction set further comprises at least:

- (iii) a command to select one of said first and second sets of registers to be an architectural set of registers accessible to said first set of functional units; and
- (iv) a command to select one of said first and second sets of registers to be an architectural set of registers accessible to said second set of functional units.

21. (Original) The embedded-DRAM processor according to Claim 20, said instruction set further comprising:

(v) a command to deselect the other of said first and second sets of registers so that it is no longer an architectural register set accessible to said first set of functional units; and

(vi) a command to deselect the other one of said first and second sets of registers so that it is no longer an architectural register set accessible to said second set of functional units.

22. (Previously Presented) The embedded-DRAM processor according to Claim 20, wherein at least one of said sets of functional units contains a single functional unit.

23. (Currently Amended) An embedded-DRAM (dynamic random access memory) processor comprising:

an embedded DRAM array comprising a plurality of random access memory cells arranged in rows and columns;

a row address register that holds a pointer that points to a row of the DRAM array; first and second registers files, each of said register files capable of ~~being loaded~~ loading or ~~stored~~ storing an entire row of the DRAM array in response to a single latch signal; and

an embedded processor comprising first and second functional units, said first and second functional units having respective first and second instruction sets and capable of accessing said first and second register files;

wherein said first and second register files comprise a parallel access port operative to parallelly transfer contents of one of said register files between a DRAM row as selected by said row-address register, said first and second register files further comprising at least a second access port operative to transfer data between a selected one of said register files and said second functional unit;

wherein said first instruction set comprises at least:

- (i) a command to manipulate data in a data register within a register file; and

wherein said second instruction set comprises at least:

- (ii) a command to perform an arithmetic operation on said row address register;
- (iii) a command to load the entire row pointed to by said row address register into a selected set of registers of said register files in a single operation.

24. (Previously Presented) The embedded-DRAM processor according to Claim 23, wherein said first and second functional units each respectively execute a command from said first and second instruction sets substantially contemporaneously.

25. (Previously Presented) The embedded-DRAM processor according to Claim 24, further comprising:

a first software module comprising data manipulation commands drawn from said first instruction set, said first software module executed by said first functional unit; and

a second software module comprising a parallel data transfer command drawn from said second instruction set, said second software module being executed by said second functional unit;

whereby said second software module operates in support of said first software module to prefetch data from said DRAM array into one of said register files in advance of said data being needed by said first software module.

26. (Previously Presented) The embedded-DRAM processor of Claim 23, wherein the second instruction set further comprises

(iv) a command to toggle a register set between an active state and an inactive state.

27. (Previously Presented) The embedded-DRAM processor of Claim 26, wherein said toggle command causes said first register file to toggle from the inactive state to the active state and also causes the second register file to toggle from the active state to the inactive state.

28. (Currently Amended) An embedded-DRAM (dynamic random access memory) processor comprising:

an embedded DRAM array comprising a plurality of random access memory cells arranged in rows and columns;

first and second dual-port registers files, each of said register files capable of parallelly transferring data between [[a]] an entire row of said DRAM array in a single operation, each of said register files also being capable of being placed into an active state and an inactive state; and

first and second embedded functional units, said first and second functional units having respective first and second instruction sets that operate on registers of an architectural register set, said architectural register set comprising one of the first and second dual-port registers files that is currently in the active state;

wherein said first instruction set comprises at least:

(i) a command to manipulate data in a data register within a register file; and

wherein said second instruction set comprises at least:

(ii) a command to unidirectionally transfer data between [[a]] an entire row of said DRAM array and a selected inactive data register file, wherein the transfer occurs in a single operation;

(iii) a command to place said selected inactive data register file into said active state, wherein when the inactive register file is activated, it becomes an architectural register set of said first functional unit.

29. (Previously Presented) The embedded-DRAM processor of Claim 28, wherein said command to unidirectionally transfer data causes data to be transferred from a row of the DRAM array to said selected inactive data register file.

30. (Previously Presented) The embedded-DRAM processor of Claim 28, wherein said command to unidirectionally transfer data causes data to be transferred from said selected inactive data register file to a row of the DRAM array.

31. (Previously Presented) The embedded-DRAM processor of Claim 28, wherein said command to place the selected inactive register file into the active state is a command that also causes the remaining register file to toggle from the active state into the inactive state.

32. (Previously Presented) The embedded-DRAM processor of Claim 28, further comprising:

at least one additional register file;

whereby said command to place the selected inactive register file into the active state is a command that also causes a selected other register file to toggle from the active state into the inactive state.

33. (Cancelled)

34. (Previously Presented) The embedded-DRAM processor of Claim 28, further comprising:

at least one bit mask; and

the second instruction set further comprises:

(iv) a command to move a subset of elements between a selected register file and a selected row of said DRAM array, whereby said subset is identified by said bit mask.

35. (Previously Presented) The embedded-DRAM processor according to Claim 28, wherein said first functional unit is a multi-issue functional unit and further comprises:

a dispatch unit;

a plurality of functional units that each execute a respective instruction stream as dispatched by said dispatch unit.

36. (Previously Presented) The embedded-DRAM processor according to Claim 28, further comprising:

a first software module comprising a set of data manipulation commands drawn from said first instruction set, said first software module executed by said first functional unit; and

a second software module comprising a set of parallel data transfer commands drawn from said second instruction set, said second software module being executed by said second functional unit;

wherein said second software module operates in support of said first software module to prefetch data from said DRAM array into one of said register tiles in advance of said data being needed by said first software module.

37. (Previously Presented) The embedded-DRAM processor according to Claim 28, wherein:

said first software module contains an instruction that references registers within an architectural register set visible to said first functional unit, whereby said architectural register set corresponds at least partially to said one of said register files that is in an active state;

said second software module contains instructions that cause data to be transferred between an inactive register set and said DRAM array, and second software module also executes a command to toggle a selected register set between said active and inactive states.

38. (Previously Presented) The embedded-DRAM processor according to Claim 28, whereby each of said register files contain a number of words, N, matched to the number of words in of a row of said DRAM array, and said unidirectional transfer comprises moving said selected row in its entirety to said selected register file.

39. (Original) The embedded-DRAM processor according to Claim 28, further comprising: a mask and switch unit interposed between said DRAM array and at least one of said register files.

40. (Previously Presented) The embedded-DRAM processor according to Claim 28, wherein said second set of instructions comprises:

a command to cause data to be moved from one register to another within a given one of said register files.

41. (Previously Presented) The embedded-DRAM processor according to Claim 28, wherein said second instruction set is used to implement an intelligent caching scheme, whereby said register files act as a cache and said second set of instructions are executed in lieu of a standard cache that maintains most recently used data and enforces a set associative or a direct-mapped caching policy.

42. (Previously Presented) The embedded-DRAM processor according to Claim 28, further comprising:

an instruction register coupled to receive instructions from said instruction set, said instruction register operative to hold an instruction to be executed by a data assembly unit; and
a local program memory coupled to said instruction register;

whereby said second functional unit corresponds to said data assembly unit, and said data assembly unit receives an instruction from said second instruction set that causes a separate control thread of instructions to be accessed from said local program memory and executed by said data assembly unit.

43. (Original) The embedded-DRAM processor of claim 42, further comprising:

a prefetch unit that prefetches instructions from the first and second instruction sets from a single very long instruction word (VLIW) instruction memory; and

a dispatch unit that dispatches instructions from the first instruction set to the functional units and dispatches instructions from the second instruction stream to the data assembly unit.

44. (Previously Presented) The embedded-DRAM processor according to Claim 28, wherein said second functional unit monitors execution activity of instructions in said first instruction set and said second instruction set further comprises:

(iv) a command to precharge a row of the DRAM array;

whereby the second functional unit executes a speculative precharging to prevent program delays due to DRAM row precharging.

45. (Currently Amended) An embedded-DRAM (dynamic random access memory) processor comprising:

an embedded DRAM array comprising a plurality of random access memory cells; first and second dual-port registers files, whereby the first port of each of said register files is a parallel access port and is parallelly coupled to said DRAM array, each of said register files being capable of being placed into an active state and an inactive state;

a functional unit that executes a first program, said functional unit coupled to said second port of said register files, said functional unit responsive to commands exclusively involving architectural register operands that map onto the registers within a register file that is in the active state;

a data assembly unit responsive to an instruction set comprising at least:

- (i) a command that causes data to be moved between the DRAM array and a register file that is in the inactive state; and
- (ii) a command that causes said register file in the inactive state to assume the active state and said register file in the active state to assume the inactive state.

46. (Currently amended) In a digital processor comprising an embedded DRAM array having a plurality of random access memory cells arranged in rows and columns, at least one row address register, the at least one row address register capable of being loaded or stored, a method comprising:

performing an arithmetic operation on said at least one row address register in order to manipulate a pointer that points to a row of the embedded DRAM array;

speculatively precharging (activating) a row pointed to by said at least one row address register based at least partially upon historical program execution data indicating a possible anticipated need to perform one or more load and or store operations that would access said row; and

in response to a separate command executed after the speculatively precharging, loading a plurality of words of a row designated by said at least one row address register into designated sets of data registers in a single operation.

47. (Previously Presented) The method of Claim 46, further comprising deactivating rows pointed to by said at least one row address register.

48. (Cancelled)

49. (Currently Amended) In a digital processor comprising an embedded DRAM array having a plurality of random access memory cells arranged in rows and columns, first and second dual-port registers files each capable of (i) parallel transfer of data between [[a]] an entire row of said embedded DRAM array in a single operation, and (ii) being placed into an active state and an inactive state, and first and second functional units, a method for processing data comprising: manipulating data in a data register within a register file that is in an the active state using said first functional unit; and

using said second functional unit;

(a) unidirectionally transferring data between [[a]] an entire row of said embedded DRAM array and a selected inactive data register file, wherein the transfer of the data occurs in a single operation; and

(b) placing said inactive register file into said active state, whereby when the register file is activated, it becomes an architectural register set of said first functional unit.

50. (Currently Amended) An embedded-DRAM (dynamic random access memory) processor comprising:

an embedded DRAM array comprising a plurality of random access memory cells arranged in rows and columns;

an embedded row address register that holds a pointer that points to a row of the embedded DRAM array;

embedded first and second registers files, each of said register files capable of ~~being loaded~~ loading or ~~stored~~ storing an entire row of the DRAM array in response to a single latch signal, each of said register files also being capable of being placed into an active state and an inactive state; and

embedded first and second of functional units, said first and second functional units having respective first and second instruction sets and capable of accessing said first and second register files while in said active state;

wherein said first and second registers files comprise a parallel access port operative to parallelly transfer contents of said register file between a DRAM row as selected by said row-address register, each of said register file further comprising at least a second access port operative to transfer data between a selected register file and said second functional unit;

wherein said first instruction set comprises at least:

a command to manipulate data in a data register within a register file; and wherein said second instruction set comprises at least:

a command to perform an arithmetic operation on said row address register; and

a command to load the entire row pointed to by said row address register into a selected set of registers of said register files that is currently in the inactive state, wherein loading the entire row is performed in a single operation.

51. (Previously Presented) The embedded DRAM processor of claim 1, wherein each of said sets of registers is capable of being loaded or stored in response to a single latch signal without a caching system that employs cache hits and cache misses.